ELSEVIER

Theoretical Computer Science 154 (1996) 307–327

Theoretical Computer Science

Effective category and measure in abstract complexity theory ☆

Cristian Calude [a,1,*], Marius Zimand [b,2]

[a] *Computer Science Department, The University of Auckland, Private Bag 92109, Auckland, New Zealand*

[b] *Department of Computer Science, University of Rochester, NY 14627, USA*

Received December 1992; revised April 1994
Communicated by M. Ito

Abstract

Strong variants of the Operator Speed-up Theorem, Operator Gap Theorem and Compression Theorem are obtained using an effective version of Baire Category Theorem. It is also shown that all complexity classes of recursive predicates have effective measure zero in the space of recursive predicates and, on the other hand, the class of predicates with almost everywhere complexity above an arbitrary recursive threshold has recursive measure one in the class of recursive predicates.

1. Introduction

The abstract complexity theory initiated by Blum [2] (see also [5, 8, 17, 23, 35]) has revealed fundamental properties of complexity measures. The striking importance of this theory relies in its machine-independent nature. Indeed, the theory is built on just two axioms (Blum axioms) and virtually any conceivable realistic model of computation is bound to satisfy the axioms. Therefore, the major achievements of this theory are primordial facts upon which any theory of complexity on any concrete model is based. The most important results of the abstract complexity theory are of existential type: (a) the existence of speedable functions [2], (b) the existence of computable functions

☆ A preliminary version of this paper, containing weaker results, has been presented to the *Second International Colloquium on Words, Languages and Combinatorics*, Kyoto Sangyo University, Japan, August 1992. Accordingly, this paper is linked to the special issue of this journal published in volume 134. An extended abstract of the paper has been presented to FCT'95, Dresden, Germany, August 1995.

* Corresponding author. Email: cristian@cs.auckland.ac.nz.

[1] The work has been supported by Auckland University Research Grants A18/XXXXX/62090/3414012, A18/XXXXX/62090/F3414022.

[2] The work has been partially supported by grants NSF-CCR-8957604, NSF-INT-9116781/JSPS-ENG-207 and NSF-CCR-9322513 and by the Romanian Department of Education and Science grant 4975-92.

SSDI 0304-3975(95)00066-6

having an arbitrary high complexity [2], (c) the existence of arbitrarily large gaps in the complexity of computable functions [4, 37], (d) the existence of a function which defines a complexity class as the union of an r.e. set of complexity classes with bounds satisfying a certain monotonicity condition (Union Theorem, [29]). The proofs of most of these theorems are extremely complex and are based on quite intricate constructions whose goal is precisely the makeup of the object whose existence is proved. Of course, one admires the crafty work and the powerful techniques involved in the proofs of these results. On the other hand, all this ingenuity spent for showing the existence of a *single* object naturally raises some questions: how typical is the existence of such an object, is it an accident or, in fact, such objects are abounding? The general mathematical practice provides two basic lines for attacking such questions: a topological one using the notion of Baire category and the measure-theoretical one. Both these approaches model in their own way the basic intuitions of fewness or largeness which are inherent to the human mind and each of them contributed with valuable insights in numerous domains of mathematics. There is one fundamental obstacle in utilizing these devices for our questions: all interesting objects in computational complexity form countable sets and, therefore, both the category analysis and the measure-theoretical one in their classical setting are much too rough to distinguish the size of such objects. Fortunately, effective versions of category and measure have been conceived that constitute natural way of investigating our problems. Due to the lack of the space, we do not go into any detail here. The definitions inside the paper provide some hints and also pointers to literature where these rather subtle matters are fully discussed. We mainly use the effective Baire category to investigate the size of the class of functions admitting an operator speed-up, the size of the class of functions defining the gap in the Operator Gap Theorem and the size of the class of functions that are almost everywhere hard to compute. This approach has already been followed in abstract complexity theory [26, 7–10, 12] and more recently in structural complexity [19, 25, 14, 40, 41]. Our results may be viewed as strenghtenings of the original basic theorems. For example, we show that the class of functions satisfying the Operator Gap Theorem is of second category. In contrast with the original version, our Speed-up Theorem cannot be deduced from the Fundamental Theorem of Complexity Theory of Meyer and Winklman [30]. In a separate section we also consider the measure-theoretical approach. Although this is an extremely active topic in structural complexity theory (see the survey of Lutz [21]), as far as we know, it is for the first time that effective measure is used in abstract complexity theory. The proofs are not deep and they closely parallel the proofs of similar results in [20, 25]; our point is that such a line of investigation is feasible and should be pursued in attacking more delicate questions in abstract complexity theory. Table 2 summarizes the currently known results and marks the contribution of this work.

2. Notation

We next describe our notations and the definitions related to the category approach. The definitions afferent to the measure-theoretical approach are deferred to Section 5.We

Table 1
The current state of affairs. Notes: (1) Definitions of objects are provided in the paper. (2) Measure refers to effective measure in the class of recursive predicates. (3) Category refers to effective Baire category. (4) "–" marks incompatibility. (5) "?" indicates an open problem

Object	Category	Measure	Where
Complexity class	I	0	Category: [26] Measure: here
a.e. complex functions	II	1	Category: here Measure: here
i.o. speedable functions	II	?	[10]
a.e. speedable functions	II	?	[12]
Operator speedable functions	II	?	here
Functions yielding gaps	II	?	[12]
Functions yielding operator gaps	II	?	here
Measured set of functions	I	0	Category: [7] Measure: here
r-honest functions	I	–	[26]

shall assume familiarity with, or access to, Bridges [5], Calude [8], Hartmanis and Hopcroft [17], Machtey and Young [23], Seiferas [35], Young [39].

Let $\mathbf{N} = \{0, 1, \ldots\}$ be the set of naturals and let $(\varphi_i)_{i\in\mathbf{N}}$ be an acceptable gödelization of **PR**, the set of unary partial recursive (p.r.) functions from **N** to **N**. Denote by **R** the class of recursive functions and by **RPRED** the class of recursive predicates, i.e. the class of functions in **R** that are $\{0,1\}$-valued. For $\varphi\in\mathbf{PR}$ we put $dom(\varphi) = \{x \in \mathbf{N} \mid \varphi(x) \text{ is defined}\}$. In what follows the term "recursive function" will always refer to a unary recursive function. Let $\langle\cdot,\cdot\rangle : \mathbf{N}^2 \to \mathbf{N}$ be a fixed pairing function. The set **FR** of p.r. functions whose domain is a finite initial segment of **N** is recursive, a fortiori recursively enumerable (r.e.) and we fix an enumeration $(\alpha_i)_{i\in\mathbf{N}}$ of **FR**.

For α in **FR**, we define the length of α by $|\alpha| = 1 + \max\{n\in\mathbf{N} \mid n\in dom(\alpha)\}$. We often consider α in **FR** as being a finite string, where the ith bit of α is $\alpha(i-1)$. We denote the set of such strings by $\mathbf{N}^{<\omega}$. If $\alpha,\beta \in \mathbf{N}^{<\omega}$, then $\alpha\beta$ denotes their concatenation. For $f,g\in\mathbf{PR}$, we write $f \sqsubseteq g$ in case $dom(f)\subseteq dom(g)$ and $f(x) = g(x)$, for every x in $dom(f)$. For every $t\in\mathbf{FR}$, put $\mathscr{U}_t \equiv \{f\in\mathbf{PR} \mid t \sqsubseteq f\}$. The family $(\mathscr{U}_t)_{t\in\mathbf{FR}}$ is a system of basic neighborhoods in **PR**; we work with the topology generated by this system (see [7, 26, 32]). In the classical framework, a set A in a topological space is *nowhere dense* (or *rare*) if for every open set $\mathcal{O}$ there exists an open subset $\mathcal{O}' \subseteq \mathcal{O}$ such that $\mathcal{O}' \cap A = \emptyset$. A set is *meager* (or of *first Baire category*) if it is a finite or denumerable union of nowhere dense sets, and it is of the *second Baire category* if it is not *meager*. In the effective variant of these notions, there exists a recursive function f which for every basic open set $\mathscr{U}_t$ produces a witness $f(t)$ which indicates the basic open set $\mathscr{U}_{f(t)}$ which is disjoint from the nowhere dense set. This ideas lead to the following definition.

Definition 2.1. (1) A set $X\subseteq\mathbf{PR}$ is *recursively nowhere dense* if there exists a recursive function f, called the *witness function*, such that:

(i) $\alpha_n \sqsubseteq \alpha_{f(n)}$, for all $n\in\mathbf{N}$,

(ii) there exists a natural $j \in \mathbf{N}$ such that for all natural $n, |\alpha_n| > j$ implies

$$X \cap \mathscr{U}_{\alpha_{f(n)}} = \emptyset.$$

(2) A set $X \subseteq \mathbf{PR}$ is *recursively meager* (or *recursively of first Baire category*) if there exist a sequence of sets $(X_i)_{i \in \mathbf{N}}$ and a recursive function f such that:

(i) $X = \bigcup_{i \in \mathbf{N}} X_i$,

and for all $i \in \mathbf{N}$:

(ii) $\alpha_n \sqsubseteq \alpha_{f(\langle i,n \rangle)}$, for all n,

(iii) there exists a natural j such that for all $n, |\alpha_n| > j$ implies

$$X_i \cap \mathscr{U}_{\alpha_{f(\langle i,n \rangle)}} = \emptyset.$$

(3) A set $X \subseteq \mathbf{PR}$ is a set of *recursively second Baire category* if X is not a set of recursively first Baire category.

For conciseness, we drop most of the times the word *recursively* in the above terminology, as well as the name of the originator of this topological classification, René Baire.

The subsets of **PR** can be classified with respect to the following hierarchy of sets of increasing size: *nowhere dense, meager, second category, co-meager and co-nowhere dense sets.* Although not all of these taxonomical notions are used in this work, we have stated them all for completeness.

One can easily observe that the extensions $\sqsubseteq$ in the above definition can be taken to be proper ($\sqsubset$) and this will be the case in all our further considerations.

The above definition can be stated in terms of the relativized topology of p.r. predicates, i.e. $\{0,1\}$-valued functions, by simply considering that $(\alpha_n)_{n \in \mathbf{N}}$ enumerates **FPRED**, the set of all p.r. predicates having the domain equal to a finite initial segment of $\mathbf{N}$. In this case, the topology is generated by the basic open sets $(\mathscr{U}_t)_{t \in \mathbf{FPRED}}$, where $\mathscr{U}_t = \{f \mid t \subset f, f \text{ is a p.r. predicate}\}$. This abuse of notation will always be clarified by context. See Calude [9] for a general treatment.

A Blum space (see [2]) is a pair $((\varphi_i)_{i \in \mathbf{N}}, (\Phi_i)_{i \in \mathbf{N}})$ where $(\varphi_i)_{i \in \mathbf{N}}$ is an acceptable gödelization of **PR** and $(\Phi_i)_{i \in \mathbf{N}}$ is a sequence of p.r. functions (called the *measure complexity* functions) satisfying the following two axioms (called *Blum axioms*): (i) $dom(\varphi_i) = dom(\Phi_i)$, for all $i \in \mathbf{N}$, and (ii) the ternary predicate $\mathbf{cost}(i,x,y) = 1$, if $\Phi_i(x) \leqslant y$, and $\mathbf{cost}(i,x,y) = 0$, otherwise, is recursive. Here, as well as in the rest of the paper, we use the following conventions. If $\Phi \in \mathbf{PR}$ and $x \in \mathbf{N}$ are such that the $\Phi(x)$ is undefined, we write $\Phi(x) = \infty$ and we consider $\infty > y$ for all $y \in \mathbf{N}$.

In the sequel we fix a Blum space $\Phi = ((\varphi_i)_{i \in \mathbf{N}}, (\Phi_i)_{i \in \mathbf{N}})$. If$g$ is a recursive function, then the set

$$C_g^{\Phi} = \{f \in \mathbf{R} \mid \text{there exists } i \text{ such that } \varphi_i = f, \Phi_i(x) < g(x) \text{ a.e. } x\}$$

is called the *complexity class* defined by g.

If $P(x,n)$ is a predicate, then we write "$P(x,n)$ a.e. n" in case $P(x,n)$ holds true for all $x \in \mathbf{N}$ and for all but a finite set of $n \in \mathbf{N}$; similarly, "$P(x,n)$ i.o. n" means that $P(x,n)$ holds true for all $x \in \mathbf{N}$ and an infinity of $n \in \mathbf{N}$.

An operator $F : \mathbf{PR} \xrightarrow{o} \mathbf{PR}$ is called *effective* if there exists a p.r. function $\psi : \mathbf{N} \xrightarrow{o} \mathbf{N}$ such that for every φ_i in the domain of F, $\psi(i)$ is defined and $F(\varphi_i)(x) = \varphi_{\psi(i)}(x)$ for every x in $\mathbf{N}$ (the notation $\xrightarrow{o}$ is used for partial mappings). The operator is *total* if it is defined on every recursive function and it preserves total recursiveness (i.e. recursive functions are mapped to recursive functions). The behaviour of effective operators is governed by the Kreisel–Lacombe–Shoenfield's Theorem. We state it in a particular form which interests total effective operators (for a proof see [8, p. 192]):

If F is a total effective operator with a recursive ψ, then there exists a recursive function $g : \mathbf{N} \to \mathbf{N}$ such that for every recursive φ_i,

$$Graph(F(\varphi_i)) = \bigcup_{j \in C_i} Graph(\alpha_{g(j)}),$$

where $C_i = \{ j \in \mathbf{N} \mid Graph(\alpha_j) \subseteq Graph(\varphi_i)\}$.

3. The speed-up phenomenon

In this section we analyze, from a topological point of view, the Operator Speed-up Theorem in its strongest form which involves p.r. predicates. Consequently, in this section we consider the topology generated by $(\mathcal{U}_t)_{t \in \mathbf{FPRED}}$. It is easy to derive all results in this section for the case of functions taking arbitrary integer values. For a total effective operator F, let $SPEED(\Phi,F)$ denote the class of recursive functions having F-speedup almost everywhere. More precisely

$$SPEED(\Phi,F) = \{ f \in \mathbf{R} \mid \text{for all } \varphi_i = f, \text{ there exists } \varphi_j = f \text{ such that } F(\Phi_j)(x) < \Phi_i(x) \text{ a.e. } x\}.$$

The main result of this section is stated in the following theorem.

Theorem 3.1. *For every total effective operator F, the set $SPEED(\Phi,F)$ is of second category.*

Proof. Fix a total effective operator F; for compactness, let denote by $SPEED$ the set $SPEED(\Phi,F)$. Assume that $SPEED$ is meager. This means that there exists a decomposition

$$SPEED = \bigcup_{j \geqslant 0} SPEED_j$$

and a recursive function f such that for every $j \geqslant 0$,

$$SPEED_j \cap U_{\alpha_{f(\langle j,n \rangle)}} = \emptyset \text{ a.e. } n.$$

We construct a recursive function g satisfying for every natural $i \geqslant 0$ the following two requirements:

$$R(i) : \alpha_{f(\langle i,m\rangle)} \subset g \text{ i.o. } m,$$

$$Q(i) : \text{ if } \Phi_i(x) < p_i(x) \text{ i.o. } x, \text{ then } g \neq \varphi_i,$$

where (p_i) is a sequence of functions built as in the standard proof of the *Operator Speed-Up Theorem* (see [28, 8]). Conditions $Q(i)$ guarantee that $g \in SPEED$. By the initial assumption, $g \in SPEED_i$, for some i. Condition $R(i)$ implies that

$$g \in \mathscr{U}_{\alpha_{f(\langle i,m\rangle)}} \text{ i.o. } m,$$

which contradicts the fact that

$$SPEED_i \cap \mathscr{U}_{\alpha_{f(\langle i,m\rangle)}} = \emptyset \text{ a.e. } m.$$

The proof proceeds by defining, in a construction by stages, a family of p.r. functions $(z_s)_{s \geqslant 0}, z_s : \mathbf{N}^4 \xrightarrow{o} \mathbf{N}$. The function we are interested in will be obtained in two steps: first we take the function z to be the limit (as s goes to ∞) of this construction (we assure that the limit exists); then, in the second step, we fix to convenient values the first three input variables of z (denoted by n, w and v) and finally obtain the desired function g. More precisely, n will be the fixed point deduced from the use of the Recursion Theorem, w will be selected in such a way as to achieve the desired rate of speed-up, and v will be chosen such that α_v will patch the initial segment of the faster program so that it exactly computes the desired function. At stage s we construct the p.r. function z_s, such that $\lambda x.z_s(n,w,v,x)$ *tries* to properly extend $\lambda x.z_{s-1}(n,w,v,x)$. It may be the case that some subcomputations in stage s cannot be performed (more precisely, the fourth condition in what is denoted *Test* below). In this case, the computation loops forever at some point in stage s. In such a situation, naturally, $\lambda x.z_t(n,w,v,x)$ is undefined for all $t \geqslant s$. However, for the value n_0 obtained through the use of the Recursion Theorem, this will not happen and, consequently, for all w, v and s, $\lambda x.z_s(n_0,w,v,x)$ properly extends $\lambda x.z_{s-1}(n_0,w,v,x)$. The particular feature of the construction is that, for all w and v, the extended part of $\lambda x.z_s(n,w,v,x)$ uses from the previous stages information related to the construction of $\lambda x.z_{s-1}(n,0,0,x)$ (and not $\lambda x.z_{s-1}(n,w,v,x)$, as one might suspect). In this way, at all stages s, we extend $\lambda x.z_s(n,w,v,x)$ for all w and v by the same amount. More precisely, we define at each stage s the integer value $Lh_s(n)$ such that for every $w, v \geqslant 0$,

$$dom(\lambda x.z_s(n,w,v,x)) = \{0,\ldots,Lh_s(n)-1\}.$$

We also use the sets $DIAG_s(n,w,v)$ with the meaning that $i \in DIAG_s(n,w,v)$ if by stage s we have insured that $z_s(n,w,v,x) \neq \varphi_i(x)$ for some $x < Lh_s(n)$. In the computation of $\lambda x.z(n,w,v,x)$ we focus at each stage s on a pair of naturals $ACTIVE_s(n,w,v) = (j,k)$, called the *active pair*, with the intention to fulfill $R(j)$. However, if at some stage we discover an index i with $i < \langle j,k\rangle$ such that $Q(i)$ can be satisfied, we prefer to do it (this is a simple form of the priority method). When $R(j)$ is satisfied, the *next* pair of naturals in a standard ordering of $\mathbf{N}^2$ becomes the new *active pair*.

The construction of $z(n,w,v,x)$

Stage $s=-1$: Put $z_{-1}(n,w,v,x)\equiv\infty$, for all $n,w,v,x\geqslant 0$, $Lh_{-1}(n)\equiv 0$, for every $n\geqslant 0$, $DIAG_{-1}(n,w,v)\equiv\emptyset$, for all $n,w,v\geqslant 0$, $ACTIVE_{-1}(n,w,v)\equiv(0,0)$.

Stage $s\geqslant 0$: Take m such that $\alpha_m=\lambda x.z_{s-1}(n,0,0,x)$ and let

$$(j,k)\equiv ACTIVE_{s-1}(n,0,0).$$

For x with $0\leqslant x<Lh_{s-1}(n)$, define $z_s(n,w,v,x)\equiv z_{s-1}(n,w,v,x)$.
for each x such that $Lh_{s-1}(n)\leqslant x<|\alpha_{f(\langle j,m\rangle)}|$ **do**
if $x\in dom(\alpha_v)$, **then** $z_s(n,w,v,x)\equiv\alpha_v(x)$
else if there exists i such that:
(1) $i<\langle j,k\rangle$,
(2) $i\notin DIAG_{s-1}(n,0,0)$,
(3) $w\leqslant i<x$,
(4) $\Phi_i(x)\leqslant\varphi_n(\langle i,x\rangle)$
Conditions (1)–(4) will be further on denoted as the *Test*.
The *Test* is checked in increasing order of i in the range
$w,w+1,\ldots,\min(x,\langle j,k\rangle)$. If for some i as above the
Test cannot be evaluated (because $\varphi_n(\langle i,x\rangle)$ is not defined)
then, of course, $z_s(n,w,v,x)$ is not defined and the procedure loops forever.
then (Satisfy one bit of $Q(i)$):
choose the least such i,
$z_s(n,w,v,x)\equiv\max\{1-\varphi_i(x),0\}$
$DIAG_{s-1}(n,w,v)\equiv DIAG_{s-1}(n,w,v)\cup\{i\}$
else (Satisfy $R(j)$)
$z_s(n,w,v,x)\equiv\alpha_{f(\langle j,m\rangle)}(x)$
end if
end for
if for all x such that $Lh_{s-1}(n)\leqslant x<|\alpha_{f(\langle j,m\rangle)}|$,
no i satisfies the *Test* or $x\in dom(\alpha_v)$), **then**
$ACTIVE_s(n,w,v)\equiv next(ACTIVE_{s-1}(n,w,v))$
else
$ACTIVE_s(n,w,v)\equiv ACTIVE_{s-1}(n,w,v)$
end if
$DIAG_s(n,w,v)\equiv DIAG_{s-1}(n,w,v)$
$Lh_s(n)\equiv|\alpha_{f(\langle j,m\rangle)}|$
End of construction

We denote $z(n,w,v,x)\equiv\lim_{s\to\infty}z_s(n,w,v,x)$ (the limit exists by the way the function z_s extends z_{s-1} for each s). Let t be a recursive function such that $\varphi_{t(n,w,v)}(x)=z(n,w,v,x)$. Now we turn to the definition of functions $(p_i)_{i\geqslant 0}$. First, let ψ be the p.r.

function defined by the following clauses:

- $\psi(n,\langle i,x\rangle) = 0$, if $x \leqslant i$ or there exists $m \leqslant i$ with $\Phi_n(\langle 0,m\rangle) \geqslant x$;
- $\psi(n,\langle i,x\rangle) = \max\{F(\Phi_{t(n,i+1,v)})(x) | v \leqslant x\}$, if the first condition fails to hold, but $\Phi_n(\langle j,y\rangle)$ is defined for all $y \leqslant x$ and all $i < j \leqslant x$;
- $\psi(n,\langle i,x\rangle) = \infty$, in the remaining situations.

By the Recursion Theorem, there exists n_0 such that $\varphi_{n_0}(u) = \psi(n_0,u)$, for all u. Fix such an n_0 and denote $p_i(x) = \varphi_{n_0}(\langle i,x\rangle)$; the complexity of $p_i(x)$ is then $\Phi_{n_0}(\langle i,x\rangle)$. One has:

$$p_i(x) = \begin{cases} 0 & \text{if } x \leqslant i \text{ or there exists } m \leqslant i \text{ with } \Phi_{n_0}(\langle 0,m\rangle) \geqslant x, \\ \max\{F(\Phi_{t(n_0,i+1,v)})(x) | v \leqslant x\} & \text{if } p_j(y) \text{ is defined for all } y \leqslant x \text{ and all } j, i < j \leqslant x, \\ \infty & \text{otherwise.} \end{cases} \tag{1}$$

We continue our proof with a series of intermediate results.

Fact 3.2. *For all natural i,x, $\varphi_{n_0}(\langle i,x\rangle)$ is defined, i.e. p_i is recursive for all i.*

Proof. We show first that $\varphi_{n_0}(\langle 0,x\rangle)$ is defined for all x, i.e. p_0 is recursive. Suppose there exists w such that $p_0(w)$ is undefined, and so is $\Phi_{n_0}(\langle 0,w\rangle)$. Hence, $\Phi_{n_0}(\langle 0,w\rangle) \geqslant x$. Then for all $i \geqslant w$, $p_i(x) = 0$ for all x, by the first clause in the definition of p_i. We retain that p_i is recursive for all $i \geqslant w$. It follows that for all v, $\lambda x.z(n_0,w,v,x)$, and thus $\varphi_{t(n_0,w,v)}$, are recursive. Indeed, this is a consequence of the fact that condition 4 in the *Test* can be effectively checked (because $\varphi_{n_0}(\langle i,x\rangle) = p_i(x)$ is defined for all $i \geqslant w$) and, if needed, $\varphi_i(x)$ can be computed in order to fulfill $Q(i)$. From (1) we notice that $p_{w-1}(x)$ is defined for all x. In the same manner, we deduce that $p_{w-2}, p_{w-3}, \ldots, p_0$ are recursive, so p_0 is recursive. Next we observe that $p_0(x)$ is defined by the second clause for almost every x, which implies that for all j, p_j is recursive. □

Fact 3.3. *For all natural v, $p_i(x) \geqslant F(\Phi_{t(n_0,i+1,v)})(x)$, a.e. x.*

Proof. Since p_i is recursive, it follows that p_i is defined by the second clause for almost every x. The conclusion follows. □

Fact 3.4. *For all natural i, there exists v such that $\varphi_{t(n_0,i,v)} = \varphi_{t(n_0,0,0)}$.*

Proof. There exists a stage s when

$$DIAG_s(n_0,0,0) \cap \{0,1,\ldots,i-1\} = \bigcup_{t \geqslant 0} DIAG_t(n_0,0,0) \cap \{0,1,\ldots,i-1\}.$$

Take v such that $\alpha_v = \lambda x.z_s(n_0,0,0,x)$. For $x < |\alpha_v|$, one has

$$\varphi_{t(n_0,i,v)}(x) = \alpha_v(x) = z_s(n_0,0,0,x) = \varphi_{t(n_0,0,0)}(x).$$

If $x \geqslant |\alpha_v|$, then $\varphi_{t(n_0,i,v)}(x) = \varphi_{t(n_0,0,0)}(x)$, by the construction of z_s. Indeed, $\varphi_{t(n_0,i,v)}$ and $\varphi_{t(n_0,0,0)}$ could differ only because at some stage t the procedures computing them satisfy $Q(h)$ and respectively $Q(h')$, with $h \neq h'$. However, after stage s, the procedure computing $\varphi_{t(n_0,0,0)}$ does not satisfy any $Q(h')$ with $h' < i$ (by the definition of s). It follows (from our strategy according to which stage s of the construction of $\lambda x.z_s(n,w,v,x)$ uses information related to $\lambda x.z_{s-1}(n,0,0,x)$, namely $ACTIVE_{s-1}(n,0,0)$ and $DIAG_{s-1}(n,0,0)$), that at each stage $t > s$, whenever a condition $Q(\cdot)$ is satisfied, the procedures computing $\varphi_{t(n_0,i,v)}$ and $\varphi_{t(n_0,0,0)}$ select the same value h for satisfying $Q(h)$. In conclusion, after stage s, the procedures computing $\varphi_{t(n_0,i,v)}$ and $\varphi_{t(n_0,0,0)}$ produce the same output, in spite of the fact that the procedure for $\varphi_{t(n_0,0,0)}$ spends exceedingly more resources (in fact, this is the source of speed-up). □

Fact 3.5. *Define the recursive function g by*

$$g(x) \equiv \varphi_{t(n_0,0,0)}(x),\ x \in \mathbf{N}.$$

For every natural j, there exist infinitely many m such that $\alpha_{f(\langle j,m\rangle)} \sqsubseteq g$, i.e. $R(j)$ is satisfied for every j.

Proof. We show first that every pair of naturals (j,k) becomes eventually the *active pair* in the algorithm computing $\varphi_{t(n_0,0,0)} = \lambda x.z(n_0,0,0,\cdot)$. We proceed by induction on $\langle j,k\rangle$. For $\langle j,k\rangle = 0$ the statement clearly holds (see stage $s = -1$). There are only finitely many naturals $i < \langle j,k\rangle$ which can satisfy the *Test* and, once such an i has satisfied the *Test*, it will not attempt doing this any more, since i is inserted in *DIAG*. Eventually no such i is found, so the construction satisfies $R(j)$ and takes the next pair of naturals as the *active pair*. Consider now the stage s when $R(j)$ is satisfied and the value m as defined at this stage. It is easy to see that

$$z_s(n_0,0,0,x) = z_{s-1}(n_0,0,0,x) = \alpha_m(x) = \alpha_{f(\langle j,m\rangle)}(x)$$

for all x with $0 \leqslant x \leqslant Lh_{s-1}(n_0)$. Moreover, the corresponding values m for the active pair (j,k) are *distinct*. Since at stage s, $R(j)$ is satisfied, we conclude that

$$g = \varphi_{t(n_0,0,0)}(\cdot) = z(n_0,0,0,\cdot) \sqsupseteq z_s(n_0,0,0,\cdot) = \alpha_{f(\langle j,m\rangle)} \text{ i.o. } m. \qquad \square$$

Fact 3.6. *If $g = \varphi_i$, then $\Phi_i(x) \geqslant p_i(x)$ a.e. x.*

Proof. There is a stage $s \geqslant 0$ such that from that moment on all active pairs (j,k) satisfy $\langle j,k\rangle > i$ and all $Q(i')$ for $i' < i$ have been satisfied. In case $i \in DIAG_{s-1}(n_0,0,0)$, φ_i was already diagonalized. Hence $\varphi_i \neq z(n_0,0,0,\cdot) = g$, a contradiction. So, $i \notin DIAG_{s-1}(n_0,0,0)$. Suppose now $\Phi_i(x) \leqslant p_i(x) = \varphi_{n_0}(\langle i,x\rangle)$, for some $x \geqslant Lh_{s-1}(n)$. There exists a stage $t \geqslant s$ such that $Lh_t(n) \geqslant x \geqslant Lh_{s-1}(n)$. It follows that at stage t the relation $z_t(n_0,0,0,\cdot) \neq \varphi_i(x)$ is realized, and hence $g = \varphi_{t(n_0,0,0,\cdot)} = z(n_0,0,0,\cdot) \neq \varphi_i$, again a contradiction. □

Proof of Theorem 3.1 (*conclusion*). We finish the proof of the theorem by showing that $g \in SPEED$. Indeed, let $\varphi_i = g$. Then $\Phi_i(x) \geqslant p_i(x) \geqslant F(\Phi_{t(n_0,i+1,v)})(x)$ a.e. x, for all $v \in \mathbf{N}$. The first inequality comes from Fact 3.6 and the second one from (1). Take v such that $\varphi_{t(n_0,i+1,v)} = \varphi_{t(n_0,0,0)}$ and $j = t(n_0, i+1, v)$. Then $\varphi_j = g$ and $\Phi_i(x) \geqslant F(\Phi_j)(x)$ a.e. x. □

Let g be an increasing recursive function and define the set

$$HARD(g) = \{f \in \mathbf{R} \mid \text{for every } \varphi_i = f,\ \Phi_i(x) > g(x) \text{ a.e. } x\},$$

of all recursive functions requiring at least $g(x)$ complexity. A well known result, due to Rabin [31] (see also Calude [8]) asserts that $HARD(g)$ is non-empty. Here we can easily derive a stronger result:

Theorem 3.7. *For every increasing recursive function g, $HARD(g)$ is of the second category.*

Proof. Consider the total effective operator F defined by $F(\varphi_i)(x) = g(x)$, for all $i, x \in \mathbf{N}$, and apply Theorem 3.1 to F, as $SPEED(\Phi, F) \subseteq HARD(\Phi, g)$. □

Note that the analogue of the above theorem for the case of polynomial-time has been established by Mayordomo [25].

It is common to derive the Speed-up Theorem (and its Operator variant) by using a *complexity sequence* and the so called Fundamental Theorem of Computational Complexity (see [30, 35, 36]). A sequence $\{p_i\}_{i \in \mathbf{N}}$ of recursive functions is a complexity sequence for a recursive function f if:

- for all i with $\varphi_i = f$, there exists j such that $p_j(x) \leqslant \Phi_i(x)$ a.e. x, and
- for all i, there exists j with $\varphi_j = f$ and $\Phi_j(x) \leqslant p_i(x)$ a.e. x.

The Fundamental Theorem states that each sequence of recursive functions satisfying some very weak conditions is the complexity sequence of some recursive function. For each total effective operator F one can find a sequence of recursive functions $\{p_i\}_{i \in \mathbf{N}}$ such that for all i, $F(p_{i+1})(x) \leqslant p_i(x)$ a.e. x. It follows immediately that the function f, for which the sequence $\{p_i\}_{i \in \mathbf{N}}$ is a complexity sequence, is F-speedable if F is non-decreasing (which is the interesting case). It is natural to ask if Theorem 3.1 (which, in fact, is a stronger version of the Operator Speed-Up Theorem, since it asserts the existence of *many* f-speedable functions), can also be obtained in this way. The answer is *negative*.

Proposition 3.8. *Let $\{p_i\}_{i \in \mathbf{N}}$ be a sequence of recursive functions. Then the set*

$$A = \{f \in \mathbf{R} \mid \{p_i\}_{i \in \mathbf{N}} \text{ is a complexity sequence for } f\}$$

is of the first category.

Proof. The set A is included in the union $\bigcup_{i,j,k\in\mathbf{N}} A_{\langle i,j,k\rangle}$, where

$$A_{\langle i,j,k\rangle} = \begin{cases} \varphi_j & \text{if } \Phi_j(x) \leqslant p_i(x), \quad \text{for all } x \text{ with } x \geqslant k\ , \\ \emptyset & \text{otherwise.} \end{cases}$$

Each set $A_{\langle i,j,k\rangle}$ is nowhere dense via the witness function $f_{\langle i,j,k\rangle}$ which on input v acts as follows:

- If $|v| < k$, then $f_{\langle i,j,k\rangle}(v) = f_{\langle i,j,k\rangle}(v0^{k-|v|})$.
- If $|v| \geqslant k$ and $\Phi_j(|v|) > p_i(|v|)$, then $f_{\langle i,j,k\rangle}(v) = v0$. (In this case $A_{\langle i,j,k\rangle} = \emptyset$ and there is nothing to worry about.)
- If $|v| \geqslant k$ and $\Phi_j(|v|) \leqslant p_i(|v|)$ then $f_{\langle i,j,k\rangle}(v) = vy$, where $y = \max\{1 - \varphi_j(|v|), 0\}$. (In this case we diagonalize over φ_j, insuring that

$$U_{f_{\langle i,j,k\rangle}(v)} \cap A_{\langle i,j,k\rangle} = \emptyset.)$$

Since the functions $f_{\langle i,j,k\rangle}$ are obtained in an uniform way from i, j, k, the conclusion follows. □

So, Theorem 3.1 *is not a corollary of the Fundamental Theorem*, as is the case of the Operator Speed-Up Theorem.

It is known that the Speed-Up Theorem is ineffective in many aspects. Blum has shown in [3] that for no speedable function φ_i can one find algorithmically from i the faster program. Although for some speedable functions it is possible to recursively bound the size of the index of the faster program (see [28, 18]), Schnorr has shown in [33] that it is not possible for any speedable function to simultaneosly bound the size of the faster program and the threshold value starting from which the faster program is indeed faster. A recent result due to Bridges and Calude [6] states that *in general, there is no recursive function of the initial index that gives a bound for the exceptional values on speed-up; but that if the bounding function is taken as a function of the speed-up index, then it can be chosen to be recursive.*

The topological analysis of the Speed-Up Theorem easily yields another facet of this phenomenon. Given any sound formal system, it is not possible to detect, with the exception of a tiny meager set of functions, that a function is speedable.

Proposition 3.9. *Let $\mathscr{T}$ be any sound formal system and F a total effective operator. There exists a function $h \in SPEED = SPEED(\Phi, F)$ such that, for each machine M computing h, the sentence* "The function computed by M belongs to $SPEED$" *is not a theorem of $\mathscr{T}$. Moreover the set of such functions h is of the second category.*

Proof. Suppose there exists a formal system $\mathscr{T}$ such that for all $h \in SPEED$ there exists a Turing machine M computing f and such that the sentence "*The function computed by M belongs to $SPEED$*" is a theorem of $\mathscr{T}$. By extracting from the theorems of $\mathscr{T}$ the ones having the above form, we obtain an r.e. sequence of machines M_i such that

$$\{h \in \mathbf{PR} \mid h \text{ is computed by some } M_i\} = SPEED.$$

The set $A_i = \{h \in \mathbf{PR} | \ h$ is computed by $M_i\}$ is nowhere dense via the witness function f_i which is defined by $f_i(v) = vy$, where $y = \max\{1 - M_i(|v|), 0\}$. It follows that *SPEED* is meager, which contradicts Theorem 3.1. The second (stronger) assertion follows also by the same reasoning. □

4. Gap and compression

A natural problem in computational complexity is to investigate to what extent by allocating more resources we get more computational power. If F is a total effective operator such that $F(f)$ is much bigger than f, are we guaranteed that

$$C_f^{\Phi} \subset C_{F(f)}^{\Phi}?$$

The Operator Gap Theorem (see [13, 8, 39]) gives a surprisingly negative result to this question: There are functions f such that $C_f^{\Phi} = C_{F(f)}^{\Phi}$. Our next result shows that there are *many* such functions.

Theorem 4.1. *Let F be a total effective operator induced by a recursive function ψ such that for all natural i and x, $F(\varphi_i)(x) \geqslant \varphi_i(x)$. Then the set*

$$GAP(\Phi, F) = \{t \in \mathbf{R} | C_t^{\Phi} = C_{F(t)}^{\Phi}\}$$

is of second category.

Proof. Suppose, by contradiction, that $GAP(\Phi, F) = \bigcup_{j \geqslant 0} GAP_j$ and there exists a recursive function f such that for all natural j:
(i) $\alpha_n \sqsubset \alpha_{f(\langle j,n \rangle)}$,
(ii) $GAP_j \cap \mathscr{U}_{\alpha_{f(\langle j,n \rangle)}} = \emptyset$,
for sufficiently large n. We construct a function $t \in GAP(\Phi, F)$ such that for all j there are infinitely many n with $\alpha_{f(\langle j,n \rangle)} \sqsubseteq t$. It will follow that for some i,

$$t \in GAP_i \cap \mathscr{U}_{\alpha_{f(\langle i,n \rangle)}} \text{ i.o. } n,$$

a contradiction.

By Kreisel–Lacombe–Shoenfield's Theorem there exists a recursive function $g : \mathbf{N} \to \mathbf{N}$ such that for every recursive φ_i,

$$Graph(F(\varphi_i)) = \bigcup_{j \in C_i} Graph(\alpha_{g(j)}),$$

where $C_i = \{j \in \mathbf{N} \mid Graph(\alpha_j) \subseteq Graph(\varphi_i)\}$.

The function t will be defined in stages. At stage s we construct a finite initial segment t_s of t and keep track of the value Lh_s such that $dom(t_s) = \{0, 1, \ldots, Lh_s - 1\}$.

Construction of t.

Stage s = 0: Put $t_0(0) \equiv 0, Lh_0 \equiv 1$.

Stage $s > 0$: Let $s \equiv \langle j,k\rangle$. (The pair (j,k) acts like the *active pair* in the previous proofs.) Let m such that $t_{s-1} = \alpha_m$. We put:

$$t_s(x) \equiv t_{s-1}(x) \quad \text{for } 0 \leqslant x < Lh_{s-1}$$

and

$$t_s(x) \equiv \alpha_{f(\langle j,m\rangle)(x)} \quad \text{for } Lh_{s-1} \leqslant x < |\alpha_{f(\langle j,m\rangle)}|.$$

(Note that we have insured that $\alpha_{f(\langle j,m\rangle)} \sqsubseteq t_s$.) Next we proceed like in Young's proof of the Operator Gap Theorem ([39, 8]). We define $s+1$ recursive extensions of the function t_s defined so far, namely $t^{(s)}, t^{(s-1)}, \dots, t^{(0)} : \mathbf{N} \to \mathbf{N}$ in the following way:

$$t^{(s)}(z) = \begin{cases} t_s(z) & \text{if } z < |\alpha_{f(\langle j,m\rangle)}|, \\ 0 & \text{otherwise,} \end{cases}$$

and for $i = s-1, s-2, \dots, 0$ (in this order)

$$t^{(i)}(z) = \begin{cases} t_s(z) & \text{if } z < |\alpha_{f(\langle j,m\rangle)}|, \\ F(t^{(i+1)})(z) & \text{otherwise.} \end{cases}$$

Next we construct the following $(x+1)$ finite initial segment functions

$$u^{(s)}, u^{(s-1)}, \dots, u^{(0)},$$

$$u^{(i)}(z) = \begin{cases} t^{(i)}(z) & \text{if } z \leqslant A_i, \\ \infty & \text{otherwise} \end{cases}$$

for every $i \in \{0,1,\dots,s\}$. Here: $A_0 = |\alpha_{f(\langle j,m\rangle)}|$ and

$$A_{i+1} = \min[z \geqslant A_i \text{ and } Graph(u^{(i)}) \subseteq \bigcup_{y \in B_{z,i}} Graph(\alpha_{g(y)})],$$

where

$$B_{z,i} = \{y \in \mathbf{N} \mid \alpha_y(n) = t^{(i+1)}(n) \quad \text{for every } n \leqslant z\}.$$

(Recall that g is the function corresponding to the operator F, by Kreisel–Lacombe–Shoenfield's Theorem.)

The idea in defining $u^{(i)}$ from $t^{(i)}$ is to retain enough information such that if $t' : \mathbf{N} \to \mathbf{N}$ is an arbitrary extension of $u^{(i)}$, then $F(t')(z) = t^{(i-1)}(z)$, for every z such that $A_0 \leqslant z \leqslant A_{i-1}$. (Observe that $F(t^{(i)})(z) = t^{(i-1)}(z)$, for every $z \geqslant A_0$.) For all $h \in \{0,1,\dots,s-1\}$, and $i \in \{0,1,\dots,s\}$, we say that $u^{(i)}$ is *unsafe* for h if

(i) $u^{(i)}(z) < \Phi_h(z)$ for some z with $Lh_{s-1} \leqslant z \leqslant A_i$,

and

(ii) $i = 0$ or $(\Phi_h(z) \leqslant u^{(i-1)}(z), i > 0)$ for all z with $Lh_{s-1} \leqslant z \leqslant A_{i-1}$.

Keeping in mind that

$$Lh_{s-1} \leqslant A_0 \leqslant A_1 \leqslant \cdots \leqslant A_s,$$

and $u^{(i)}(z) \leqslant u^{(i-1)}(z)$, for all z with $Lh_{s-1} \leqslant z \leqslant A_{i-1}$, for all $i \in \{1,2,\dots,s\}$, one deduce that for every $h \in \{0,1,\dots,s-1\}$ at most one $u^{(i)}$ is *unsafe* for h. Since there are $s+1$

such extensions $u^{(i)}$, at least one of them is *safe* for all $h < s$, and such an $u^{(i)}$ can be found in a recursive way. We extend t_s to a certain $u^{(i_0)}$ which is *safe* for every $h < s$ and set $Lh_s \equiv A_{i_0} + 1$.

End of stage s.

Finally, let $t \equiv \lim_{s\to\infty} t_s$. *End of construction of t.*

For each i there are infinitely many k such that $\alpha_{f(\langle i,k\rangle)} \sqsubseteq t$, since at each step $s = \langle i,k\rangle$, we make $\alpha_{f(\langle i,k\rangle)} \sqsubseteq t_s$ and $t_s \sqsubseteq t$. It remains to show that $t \in GAP(\Phi,F)$. Suppose there exists j such that $\Phi_j(x) \leqslant F(t)(x)$ a.e. x, but $\Phi_j(x) > t(x)$ i.o. x. There exists a stage $s > j$ such that $\Phi_j(x) \leqslant F(t)(x)$ for all x with $Lh_{s-1} \leqslant x < Lh_s$ and $\Phi_j(x) > t(x)$ for some x with $Lh_{s-1} \leqslant x < Lh_s$. This contradicts the choice of a *safe* initial segment at stage s. Indeed, if $Lh_{s-1} \leqslant x < Lh_s$, then $t(x) = u^{(i_0)}(x)$, where $u^{(i_0)}$ is the *safe* segment selected at stage s. If $i_0 = 0$, then $u^{(i_0)}$ is a *safe* segment if and only if $u^{(i_0)}(x) \geqslant \Phi_j(x)$, for all x with $Lh_{s-1} \leqslant z \leqslant A_0 = Lh_s - 1$. If $i_0 > 0$, then for all $x \leqslant A_{i_0-1}, F(t)(x) = F(u^{(i_0)})(x) = u^{(i_0-1)}(x)$ and $A_{i_0-1} \geqslant Lh_{s-1}$. So our assumptions on Φ_j would imply that $u^{(i_0)}$ is not *safe*. □

The Compression Theorem [2] assures us that for *nice* families of functions T, one can indeed get more computational power by raising the resource bound from $t(x)$ to $g(x, t(x))$, for an appropriate g and for any $t \in T$. *Nice* means here a *measured* set, i.e. an r.e. set of partial functions $\gamma_i : \mathbf{N} \xrightarrow{o} \mathbf{N}$ for which the ternary predicate $\gamma_i(n) = m$ is recursive. The Compression Theorem states that if $(\gamma_i)_{i\in\mathbf{N}}$ is a measured set, then there exist two recursive function $g: \mathbf{N}^2 \to \mathbf{N}$ and $k: \mathbf{N} \to \mathbf{N}$ such that for each recursive γ_i:

(i) if $\varphi_j = \varphi_{k(i)}$, then $\Phi_j(x) > \gamma_i(x)$ a.e. x and

(ii) $\Phi_{k(i)}(x) \leqslant g(x, \gamma_i(x))$ a.e. x.

We would like to know what properties should a set of functions have in order to possess a Compression Theorem (being a measured set is probably not a necessary condition). It is known from Calude [7] (see also [8]) that every measured set is meager. (In [7], the result is shown in a different topology, the *superset* topology, which is adequate for investigating the size of sets of partial recursive functions. However the proof from [7] can easily be adapted for the Cantor topology.) Our next result shows that the meagerness of a measured set seems to be essential and gives a partial answer to the above question. If we fix the increasing factor g and consider a second category set A of recursive functions, then the compression property does not hold for any f in A.

Proposition 4.2. *Let $g : \mathbf{N}^2 \to \mathbf{N}$ be a recursive function. Then the set*

$$A_g = \{s \in \mathbf{R} \mid \text{there exists } i \text{ with } \Phi_i(x) \leqslant g(x, s(x)) \text{ a.e. } x \text{ and for all } j \text{ with } \varphi_j = \varphi_i \text{ one has } \Phi_j(x) > s(x) \text{ a.e. } x\}$$

is meager.

Proof. We start noticing that

$$A_g \subseteq \bigcup_{i,k \geqslant 0} A_{\langle i,k\rangle},$$

where

$$A_{\langle i,k\rangle} = \{s \in \mathbf{R} \mid \Phi_i(x) \leqslant g(x,s(x)) \text{ and } \Phi_i(x) > s(x) \text{ for all } x \geqslant k\}.$$

The set $A_{\langle i,k\rangle}$ is nowhere dense via the following witness function $f_{\langle i,k\rangle} : \mathbf{N}^{<\omega} \to \mathbf{N}^{<\omega}$:

1. If $v \in \mathbf{N}^{<\omega}$ has length less than k then we (arbitrarily) put $f_{\langle i,k\rangle}(v) = v0$.
2. If $|v| \geqslant k$, then let $x = |v|$.

- If $\Phi_i(x) \leqslant g(x,0)$ put $f_{\langle i,k\rangle}(v) = vy$, where $y = \Phi_i(x) + 1$. (In this way, if $f_{\langle i,k\rangle}(v)$ is a prefix of s, then $\Phi_i(x) \leqslant s(x)$.)
- If $\Phi_i(x) > g(x,0)$ put $f_{\langle i,k\rangle}(v) = v0$. (In this way, if $f_{\langle i,k\rangle}(v)$ is a prefix of s, then $\Phi_i(x) > g(x,s(x))$.)

It is clear from the above remarks that for any $v \in \mathbf{N}^{<\omega}$ with $|v| \geqslant k$,

$$U_{f_{\langle i,k\rangle}(v)} \cap A_{\langle i,k\rangle} = \emptyset,$$

and, consequently, A_g is meager. □

It is not hard to prove that any function in a measured set is r-honest, for an appropriate recursive function r. (We recall that, in a Blum space (φ_i, Φ_i), a recursive function f is r-honest if there is an i such that $\varphi_i = f$ and $\Phi_i(x) \leqslant r(x, f(x))$ a.e. x; here $r : \mathbf{N}^2 \to \mathbf{N}$ is a recursive function.) Consequently, the functions in the Compression Theorem are all honest. It is also well known that the classical time and space hierarchy theorems for Turing machines (see [22]) require time and, respectively, space constructibility of the involved functions, which is a strong variant of honesty. One may suspect that functions defining the gaps in the "anti-hierarchy" Operator Gap Theorem are not honest. Indeed, we can prove that many of them are not honest in an extremely strong sense. Fix a *uniform* sequence of Blum spaces $(\varphi^{(i)}, \Phi^{(i)})_{i \in \mathbf{N}}$, in the sense that there exists a recursive function $V : \mathbf{N}^2 \to \mathbf{N}$ such that $V(i,x) = \Phi^{(i)}(x)$ for all i and x.

Proposition 4.3. *Let r, g be recursive functions and $(\varphi^{(i)}, \Phi^{(i)})$ as above. Then the set of functions in $GAP(\Phi, g)$ which are not r-honest in any Blum space $(\varphi^{(i)}, \Phi^{(i)})$ is of second category.*

Proof. By a result of Mehlhorn [26], the class of r-honest functions is meager. It follows that the set

$$A = \{f \in \mathbf{PR} \mid f \text{ is } r\text{-honest in } (\varphi^{(i)}, \Phi^{(i)}), \text{ for some natural } i\}$$

is also meager, since it is the uniform union of a countable set of meager sets. Since, by Theorem 4.1, $GAP(\Phi, g)$ is of second category, we conclude that $GAP(\Phi, g) \setminus A$ is of second category as well. □

5. Effective measure

To the best of our knowledge, there has been no prior investigation of the size of objects in abstract computational complexity from an effective measure theoretical point of view. It is the aim of this section to illustrate that such a study is quite feasible. In fact, we easily classify from the effective measure-theoretical point of view all classes of predicates that were analyzed in the topological setting in the early works of Mehlhorn [26] and Calude [7]. Similar results to the ones below have been established for the case of natural complexity measures in [20, 25].

The recursive and resource-bounded variants of measure theory have been developed by Freidzon [15], Mehlhorn [27], Schnorr [34] and exhaustively by Lutz [20]. It is important to retain that this theory is applicable only to $\{0,1\}$-valued functions, i.e. to predicates (and this is a major drawback of effective measure when compared to effective category; for example, the size of the class of r-honest functions cannot be analyzed in the measure-theoretical setting). The following definitions come from the latter paper, restricted to the necessities of the current work. For motivations and connections with classical measure theory, we direct the reader to [20, 21, 1, 25]. Let Σ^* and Σ^∞ be the sets of finite and, respectively, infinite binary strings and let λ denote the empty string. If $w \in \Sigma^*$, then w can be identified with a function in **FPRED** which for all $i < |w|$ maps i into the $(i+1)$th bit of w. The basic neighborhoods $\mathscr{U}_w$ are defined as in Section 1 (in the topology of p.r. predicates).

Definition 5.1. (i) A *density function* is a function $d : \Sigma^* \to [0,\infty)$ satisfying

$$d(w) \geqslant \frac{d(w0)+d(w1)}{2}$$

for all strings $w \in \Sigma^*$.

(ii) The *global value* of a density function d is $d(\lambda)$. The set covered by a density function d is

$$S[d] = \bigcup_{w \in \Sigma^*, d(w) \geqslant 1} \mathscr{U}_w.$$

A density function d covers a set $X \subseteq \Sigma^\infty$ if $X \subseteq S[d]$.

(iii) An *1-dimensional density system* is a function $d : \mathbf{N} \times \Sigma^* \to [0,\infty)$ such that for all $i \in \mathbf{N}$, d_i is a density function, where d_i is defined by $d_i(x) = d(i,x)$.

Definition 5.2. (i) A set $X \subseteq \Sigma^\infty$ has *recursive measure zero* if there exists an 1-dimensional recursive density system d (i.e. d is a recursive real-valued function) such that for all k, d_k covers X with global value $d_k(\lambda) \leqslant 2^{-k}$.

(ii) A set $X \subseteq \Sigma^\infty$ has *recursive measure one* if the complement of X has recursive measure zero.

The general idea is to give an effective touch to the standard method in measure theory consisting in the covering of sets by intervals. This is realized by using recursive martingales (called here density functions) betting on the cylinders $\mathscr{U}_w$. The initial investment is the global value. Thus, roughly speaking, a set has recursive measure zero, if there is a recursive winning betting strategy covering the set that starts with an arbitrary small amount of money. In fact, one could weaken condition (i) in Definition 5.2 by requiring that the density system has only arbitrarily close recursive approximations, but, practically, in all proofs, the density system is recursive itself.

A function $f \in$ **RPRED** will be identified with the infinite binary string $f(0)f(1)\ldots \in \Sigma^\infty$. Lutz [20] has shown that **RPRED** *does not have recursive measure zero.* This important feature makes a recursive measure meaningful for studying the size of classes of functions in **RPRED** with respect to the whole space **RPRED**.

Definition 5.3. (i) A set $X \subseteq \Sigma^\infty$ has *recursive measure zero* in **RPRED** if $X \cap$ **RPRED** has recursive measure zero. (ii) A set $X \subseteq \Sigma^\infty$ has *recursive measure one* in **RPRED** if the complement of X has recursive measure zero in **RPRED**.

The following useful result is proved by Lutz [20]: *A recursive union of recursive measure zero sets has recursive measure zero.* Formally, let $d : N^2 \times \Sigma^* \to [0,\infty)$ be a recursive function such that for each j, d_j is a density system (where $d_j(i,x) = d(j,i,x)$, for all $i\in\mathbf{N}$ and $x\in\Sigma^*$). If $X, X_1, X_2, \ldots \subseteq \Sigma^\infty$, $X = \bigcup_{i=0}^{\infty} X_i$ and for all $i\in\mathbf{N}$, d_i is a witness that X_i has recursive measure zero, then X has recursive measure zero. The following is the analogous result from the point of view of recursive measure theory of Theorem 3.7 (and surprisingly the proof is much simpler).

Theorem 5.4. *For every recursive function g, $Hard(g) \cap$* **RPRED** *has recursive measure one in* **RPRED**.

Proof. We have to show that the set

$$HARD^c(g) = \{f \in RPRED \mid \text{there exists } \varphi_i = f, \Phi_i(x) \leqslant g(x) \text{ i. o. } x\}$$

has recursive measure zero. Let $((\varphi_i)_{i\in\mathbf{N}}, (\Phi_i)_{i\in\mathbf{N}})$ be a Blum space of $\{0,1\}$-valued p.r. functions. It is immediate that $HARD^c(g) \subseteq \bigcup H_i$, where

$$H_i = \begin{cases} \{f \in RPRED \mid \varphi_i = f\} & \text{if there exist infinitely many } x \\ & \text{such that } \Phi_i(x) \leqslant g(x), \\ \emptyset & \text{otherwise.} \end{cases}$$

We describe a recursive function $d : \mathbf{N}^2 \times \Sigma^* \to [0,\infty)$ such that d_i witnesses that H_i has effectively measure zero, where $d_i(k,x) = d(i,k,x)$, for all $i \in \mathbf{N}$, and $x \in \Sigma^*$. In fact, we define the "projections" $d_{i,k}$ as follows:

$$d_{i,k}(\lambda) = 2^{-k},$$

and, inductively,

$$d_{i,k}(x0) = \begin{cases} 0 & \text{if } \Phi_i(|x|) \leqslant g(|x|) \text{ and } \varphi_i(|x|) = 1, \\ 2d_{i,k}(x) & \text{if } \Phi_i(|x|) \leqslant g(|x|) \text{ and } \varphi_i(|x|) = 0, \\ d_{i,k}(x) & \text{otherwise,} \end{cases}$$

$$d_{i,k}(x1) = \begin{cases} 2d_{i,k}(x) & \text{if } \Phi_i(|x|) \leqslant g(|x|) \text{ and } \varphi_i(|x|) = 1, \\ 0 & \text{if } \Phi_i(|x|) \leqslant g(|x|) \text{ and } \varphi_i(|x|) = 0, \\ d_{i,k}(x) & \text{otherwise.} \end{cases}$$

It can readily be checked that $d_{i,k}$ is a density function. Also, by induction on h, one can see that $d_{i,k}(x) \geqslant 2^{h-k}$ if and only if there exist more than h integers y with $y \leqslant |x| - 1$ and $\Phi_i(y) < g(y)$. Therefore, it follows that

$$H_i \subseteq S[d_{i,k}]$$

for all $k \in \mathbf{N}$. □

The above proof also implies the effective measure-theoretical analogue of the basic result of Mehlhorn [26], Calude [7] stating that *all complexity classes are meager*. Indeed, if $((\varphi_i)_{i\in\mathbf{N}}, (\Phi_i)_{i\in\mathbf{N}})$ is a Blum space of $\{0,1\}$-valued p.r. functions, and g is a recursive function, then the complexity class defined by g, C_g^{Φ}, is included in $HARD^c(g)$, the set defined in the above proof and shown to have effective measure zero in **RPRED**. Therefore:

Theorem 5.5. *Let $((\varphi_i)_{i\in\mathbf{N}}, (\Phi_i)_{i\in\mathbf{N}})$ be a Blum space of $\{0,1\}$-valued p.r. functions and g a recursive function. Then C_g^{Φ} has effective measure zero in* **RPRED**.

We pass to the study of measured set of predicates (see the definition in the previous section). We are interested in measured sets of recursive predicates, i.e. sets of recursive $\{0,1\}$-valued functions γ_i. Many natural classes of recursive predicates form measured sets, e.g. the class of primitive recursive predicates, every r.e. complexity class of predicates and, in fact, any r.e. set of recursive predicates. It follows from the next theorem that all these classes (and all their subclasses, like all levels in Grzegorczyk's hierarchy) have effective measure zero.

Theorem 5.6. *If $\Gamma = (\gamma_i)_{i\in\mathbf{N}}$ is a measured set of recursive predicates, then Γ has effective measure zero.*

Proof. We decompose $\Gamma = \bigcup \Gamma_i$, where $\Gamma_i = \{\gamma_i\}$. It is immediate to build, for each i, a density system d_i witnessing that Γ_i has recursive measure zero. Namely, $d_{i,k}(\lambda) = 2^{-k}$, and

$$d_{i,k}(x0) = \begin{cases} 0 & \text{if } \gamma_i(|x|) = 1, \\ 2d_{i,k}(x) & \text{if } \gamma_i(|x|) = 0, \end{cases} \quad d_{i,k}(x1) = \begin{cases} 2d_{i,k}(x) & \text{if } \gamma_i(|x|) = 1, \\ 0 & \text{if } \gamma_i(|x|) = 0. \end{cases} \quad □$$

6. Final discussion

The "?" marked entries in Table 2 constitute as many interesting open questions. The status of the class of non-speeadable functions remains *open*. It seems that the exact Baire classification of the set of non-F-speedable functions depends upon F. We believe that if F is a fast growing function, then the corresponding set of non-F-speedable functions might be of second category. This claim can be proved, for instance, in case we restrict the class of witness functions to an r.e. class of recursive functions. All our results involved Cantor's topology which is based on extension of initial finite segments of functions. With some care, the proofs can be adapted to work for the case of the *superset* topology, which is based on extension of finite sets (see [8]). It would be interesting to find the most general class of topologies for which our results hold. Such an analysis has been pursuited for the case of independent statements in a formal theory in [11].

It is interesting to note, folowing Marcus [24], that, in real analysis, the property of meagerness of a class of functions is – to some extent – conditioned by the degree of effectiveness of the definition of functions. This situation extends over all previous studied sets in abstract complexity theory and includes most of the present results. Fulk [16] has shown that every proof of the existence of almost everywhere arbitrarily complex predicates should have a non-constructive component. The proof is partially constructive in that it effectively gives a program for the function; the non-constructive argument is used for the verification that the constructed function has the desired properties. In case of speedable functions the situation is, from the constructive point of view, worse: the construction of the function itself uses a sequence of programs from which the desired function is selected noneffectively (see [2]). The natural question is whether there is a more profound connection between non-effectiveness and the property of being topologically large.

Acknowledgements

The authors thank the anonymous referee whose valuable comments led them to get stronger results.

References

[1] E. Allender and M. Strauss, Measure on small complexity classes, with applications for BPP, *FOCS'94* (1994) 807–818.
[2] M. Blum, A machine-independent theory of the complexity of recursive functions, *J. ACM* **14**(2) (1967) 322–336.
[3] M. Blum, On effective procedures for speeding up algorithms, *J. ACM* **18**(2) (1967) 257–265.
[4] A. Borodin, Computational complexity and the existence of complexity gaps, *J. ACM* **19**(1) (1972) 158–174.
[5] D.S. Bridges, *Computability – A Mathematical Sketchbook* (Springer, Berlin, 1994).

[6] D.S. Bridges and C. Calude, On recursive bounds for the exceptional values in speed-up, *Theoret. Comput. Sci.* **132** (1994) 387-394.

[7] C. Calude, Topological size of sets of partial recursive functions, *Z. Math. Logik Grundlag. Math.* **28**(1982) 455–462.

[8] C. Calude, *Theories of Computational Complexity* (North-Holland, Amsterdam, 1988).

[9] C. Calude, Relativized topological size of sets of partial recursive functions, *Theoret. Comput. Sci.* **87** (1991) 347–352.

[10] C. Calude, G. Istrate and M. Zimand, Recursive Baire classification and speedable functions, *Z. Math. Logik Grundlag. Math.* **3** (1992) 169–178.

[11] C. Calude, H. Jürgensen and M. Zimand, Is independence an exception?, *Appl. Math. Comput.* **66** (1994) 63–76.

[12] C. Calude and M. Zimand, On three theorems in abstract complexity theory: A topological glimpse, *Abstracts of the Second International Colloquium on Semigroups, Formal Languages and Combinatorics on Words*, Kyoto, Japan (1992) 11–12.

[13] R.L. Constable, The operator gap, *J. ACM* **19**(1) (1972) 175–183.

[14] S. Fenner, Notions of resource-bounded category and genericity, *Proc. 6th Structure in Complexity Theory* (1991) 347–352.

[15] R. Freidzon, Families of recursive predicates of measure zero, *J. Soviet Math.* **6** (1976) 449–455.

[16] M.A. Fulk, A note on a.e. h-complex functions, *J. Comput. System Sci.* **40** (1990) 444–449.

[17] J. Hartmanis and J.E. Hopcroft, An overview of the theory of computational complexity, *J. ACM* **18**(3) (1971) 444–475.

[18] J. Helm and P. Young, On size vs. efficiency for programs admitting speed-ups, *J. Symbolic Logic* **36** (1971) 21–27.

[19] J. Lutz, Category and measure in complexity theory, *SIAM J. Comput.* **19** (1990) 1100–1131.

[20] J. Lutz, Almost everywhere high nonuniform complexity, *J. Comput. System Sci.* **44** (1992) 220–258.

[21] J. Lutz, The quantitative structure of exponential time, *Proceedings of the 8th Structure in Complexity Theory Conference* (1993) 158–175.

[22] J.E. Hopcroft and J.D. Ullman, *An Introduction to Automata Theory, Languages and Computation* (Addison-Wesley, Reading, MA, 1979).

[23] M. Machtey and P. Young, *An Introduction to the General Theory of Algorithms* (North-Holland, Amsterdam, 1978).

[24] S. Marcus, Personal Communication, July 1993.

[25] E. Mayordomo, Almost every set in exponential time is p-bi-immune, *Theoret. Comput. Sci.* **136** (1994) 487–506.

[26] K. Mehlhorn, On the size of sets of computable functions, *Annual IEEE Symp. on Switching and Automata Theory*, Univ. Iowa, Ames, IA (1973) 190–196.

[27] K. Melhorn, The almost all theory of subrecursive degrees is decidable, *Proc. Second ICALP*, Lecture Notes in Computer Science (Springer, Berlin, 1974) 1317–325.

[28] A.R. Meyer and P.C. Fischer, Computational speed-up by effective operators, *J. Symbolic Logic* **37**(1) (1972) 55–68.

[29] E. McCreight and A. Meyer, Classes of computable functions defined by bounds on computation: preliminary report, *Conf. Rec. ACM Symp. on Theory of Computing* (1965) 79–88.

[30] A.R. Meyer and K. Winklman, The fundamental theorem of complexity theory, in J.W. de Bakker and J. van Leeuwen, eds., *Found. Comput. Sci. III Part 1: Automata, Data Structures, Complexity*, Vol. 108 (Mathematical Centre Tracts, Amsterdam, 1979) 97–112.

[31] M. Rabin, Degree of difficulty of computing a function, Hebrew University, Jerusalem, Technical Report 2 (April 25), 1960.

[32] H. Rogers, *Theory of Recursive Functions and Effective Computability* (McGraw-Hill, New York, 1967).

[33] C.P. Schnorr, Does the computational speed-up concern programming?, *Proc. First Internat. Conf. on Automata, Languages and Programming* (1972) 589–596.

[34] C.P. Schnorr, Process complexity and effective random tests, *J. Comput. System Sci.* **7** (1973) 376–388.

[35] J. Seiferas, Machine-independent complexity theory, in J. van Leeuwen, ed., *Handbook of Theoretical Computer Science, Vol. A* (Elsevier, Amsterdam, 1990) 165–186.

[36] J. Seiferas and A.R. Meyer, Characterization of realizable space complexities, *Ann. Pure Appl. Logic* **73** (1995) 171–190.

[37] B.A. Trakhtenbrot, *Complexity of Algorithms and Computations*, Course Notes, Novosibirsk, 1967 (Russian).

[38] P. van Emde Boas, Ten years of speed-up, *Proc. of the Symp. on Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science, Vol. 32 (Springer, Berlin, 1975) 232–237.

[39] P. Young, Easy constructions in complexity theory: gap and speed-up theorems, *Proc. Amer. Math. Soc.* **37** (1973) 555–563.

[40] M. Zimand, If not empty, NP\P is topologically large, *Theoret. Comput. Sci.* **119** (1993) 293–310.

[41] M. Zimand, On the topological size of p-m-complete degrees, *Theoret. Comput. Sci.* **147** (1995) 137–147.